1 SCALABLE HIERARCHICAL DATA-DRIVEN NAVIGATION SYSTEM AND

2 METHOD FOR INFORMATION RETRIEVAL

3 This application is a continuation-in-part of Application Ser. No. 09/573,305,

4 entitled "Hierarchical Data-Driven Navigation System and Method for Information

5 Retrieval," filed May 18, 2000, and incorporated herein by this reference.

6 1. Field of the Invention

7 The present invention generally relates to information navigation systems and

8 search engines.

9 2. Background of the Invention

10 Information retrieval from a database of information is an increasingly

11 challenging problem, particularly on the World Wide Web (WWW), as increased

12 computing power and networking infrastructure allow the aggregation of large amounts

13 of information and widespread access to that information. A goal of the information

14 retrieval process is to allow the identification of materials of interest to users.

15 As the number of materials that users may search increases, identifying materials

16 relevant to the search becomes increasingly important, but also increasingly difficult.

17 Challenges posed by the information retrieval process include providing an intuitive,

18 flexible user interface and completely and accurately identifying materials relevant to the

19 user's needs within a reasonable amount of time. Another challenge is to provide an

20 implementation of this user interface that is highly scalable, so that it can readily be

21 applied to the increasing amounts of information and demands to access that information.

1   The information retrieval process comprehends two interrelated technical aspects,

2   namely, information organization and access.

3   Current information navigation systems usually follow one of three paradigms.

4   One type of information navigation system employs a database query system. In a typical

5   database query system, a user formulates a structured query by specifying values for

6   fixed data fields, and the system enumerates the documents whose data fields contain

7   those values. PriceSCAN.com uses such an interface, for example. Generally, a database

8   query system presents users with a form-based interface, converts the form input into a

9   query in a formal database language, such as SQL, and then executes the query on a

10   relational database management system. Disadvantages of typical query-based systems

11   include that they allow users to make queries that return no documents and that they offer

12   query modification options that lead only to further restriction of the result set (the

13   documents that correspond to the user's search specifications), rather than to expansion or

14   extension of the result set. In addition, database query systems typically exhibit poor

15   performance for large data sets or heavy access loads; they are often optimized for

16   processing transactions rather than queries.

17   A second type of information navigation system is a free-text search engine. In a

18   typical free-text search engine, the user enters an arbitrary text string, often in the form of

19   a Boolean expression, and the system responds by enumerating the documents that

20   contain matching text. Google.com, for example, includes a free-text search engine.

21   Generally a free-text search engine presents users with a search form, often a single line,

22   and processes queries using a precomputed index. Generally this index associates each

1   document with a large portion of the words contained in that document, without

2   substantive consideration of the document's content. Accordingly, the result set is often

3   a voluminous, disorganized list that mixes relevant and irrelevant documents. Although

4   variations have been developed that attempt to determine the objective of the user's query

5   and to provide relevance rankings to the result set or to otherwise narrow or organize the

6   result set, these systems are limited and unreliable in achieving these objectives.

7   A third type of information navigation system is a tree-based directory. In a tree-

8   based directory, the user generally starts at the root node of the tree and specifies a query

9   by successively selecting refining branches that lead to other nodes in the tree.

10   Shopping.yahoo.com uses a tree-based directory, for example. In a typical

11   implementation, the hard-coded tree is stored in a data structure, and the same or another

12   data structure maps documents to the node or nodes of the tree where they are located. A

13   particular document is typically accessible from only one or, at most, a few, paths

14   through the tree. The collection of navigation states is relatively static—while documents

15   are commonly added to nodes in the directory, the structure of the directory typically

16   remains the same. In a pure tree-based directory, the directory nodes are arranged such

17   that there is a single root node from which all users start, and every other directory node

18   can only be reached via a unique sequence of branches that the user selects from the root

19   node. Such a directory imposes the limitation that the branches of the tree must be

20   navigationally disjoint—even though the way that documents are assigned to the disjoint

21   branches may not be intuitive to users. It is possible to address this rigidity by adding

22   additional links to convert the tree to a directed acyclic graph. Updating the directory

1 structure remains a difficult task, and leaf nodes are especially prone to end up with large

2 numbers of corresponding documents.

3 In all of these types of navigation systems, it may be difficult for a user to revise a

4 query effectively after viewing its result set. In a database query system, users can add or

5 remove terms from the query, but it is generally difficult for users to avoid underspecified

6 queries (i.e. too many results) or overspecified queries (i.e. no results). The same

7 problem arises in free-text search engines. In tree-based directories, the only means for

8 users to revise a query is either to narrow it by selecting a branch or to generalize it by

9 backing up to a previous branch.

10 Having an effective means of revising queries is useful in part because users often

11 do not know exactly what they are looking for. Even users who do know what they are

12 looking for may not be able to express their search criteria precisely. And the state of the

13 art in search technology cannot guarantee that even a precisely stated query will be

14 interpreted as intended by the user. Indeed, it is unlikely that a perfect means for

15 formation of a query even exists in theory. As a result, it is helpful that the information

16 retrieval process be a dialogue with interactive responses between the user and the

17 information retrieval system. This dialogue model may be more effectively implemented

18 with an effective query revision process.

19 Various other systems for information retrieval are also available. For example.

20 U.S. Patents Nos. 5,715,444 and 5,983,219 to Danish et al., both entitled "Method and

21 System for Executing a Guided Parametric Search," disclose an interface for identifying a

22 single item from a family of items. The interface provides users with a set of lists of

1   features present in the family of items and identifies items that satisfy selected features.

2   Other navigation systems include i411's Discovery Engine, Cybrant's Information

3   Engine, Mercado's IntuiFind, and Requisite Technology's BugsEye.

4   3. Summary of the Invention

5   The present invention, a highly scalable, hierarchical, data-driven information

6   navigation system and method, enables the navigation of a collection of documents or

7   other materials using certain common attributes associated with those materials. The

8   navigation system interface allows the user to select values for the attributes associated

9   with the materials in the current navigation state and returns the materials that correspond

10   to the user's selections. In some embodiments, the user's selections may be constrained

11   using Boolean operators. The present invention enables this navigation mode by

12   associating terms (attribute-value pairs) with the documents, defining a set of hierarchical

13   refinement relationships (i.e., a partial order) among the terms, and providing a guided

14   navigation mechanism based on the association of terms with documents and the

15   relationships among the terms.

16   The present invention includes several components and features relating to a

17   hierarchical data-driven navigation system. Among these are a user interface, a

18   knowledge base, a process for generating and maintaining the knowledge base, a

19   navigable data structure and method for generating the data structure, WWW-based

20   applications of the system, and methods of implementing the system. Although the

21   invention is described herein primarily with reference to a WWW-based system for

22   navigating a product database, it should be understood that a similar navigation system

1   could be employed in any database context where materials may be associated with terms

2   and users can identify materials of interest by way of those terms.

3   The present invention uses a knowledge base of information regarding the

4   collection of materials to formulate and to adapt the interface to guide the user through

5   the collection of navigation states by providing relevant navigation options. The

6   knowledge base includes an enumeration of attributes relevant to the materials, a range of

7   values for each attribute, and a representation of the partial order that relates terms (the

8   attribute-value pairs). Attribute-value pairs for materials relating to entertainment, for

9   example, may be *Products: Movies* and *Director: Spike Lee*. (Attribute-value pairs are

10   represented throughout this specification in this *Attribute: Value* format; navigation

11   states are represented as bracketed expressions of attribute-value pairs.) The knowledge

12   base also includes a classification mapping that associates each item in the collection of

13   materials with a set of terms that characterize that item.

14   The knowledge base is typically organized by domains, which are sets of

15   materials that conform to natural groupings. Preferably, a domain is chosen such that a

16   manageable number of attributes suffice to effectively distinguish and to navigate among

17   the materials in that domain. The knowledge base preferably includes a characterization

18   of each domain, which might include rules or default expectations concerning the

19   classification of documents in that domain. A particular item may be in more than one

20   domain.

21   The present invention includes a user interface for navigation. The user interface

22   preferably presents the user's navigation state as an expression of terms organized by

1 attribute. For a given expression of terms, the user interface presents materials that are

2 associated with those terms in accordance with that expression and presents relevant

3 navigation options for narrowing or for generalizing the navigation state. In one aspect

4 of the present invention, users navigate through the collection of materials by selecting

5 and deselecting terms.

6   In one aspect of the present invention, the user interface responds immediately to

7 the selection or the deselection of terms, rather than waiting for the user to construct and

8 to submit a comprehensive query composed of multiple terms. Once a query has been

9 executed, the user may narrow the navigation state by conjunctively selecting additional

10 terms, or by refining existing terms. Alternatively, the user may broaden the navigation

11 state by deselecting terms that have already been conjunctively selected or by

12 generalizing the terms. In preferred embodiments, the user may broaden the navigation

13 state by deselecting terms in an order different from that in which they were

14 conjunctively selected. For example, a user could start at {*Products: Movies*}, narrow by

15 conjunctively selecting an additional term to {*Products: Movies* AND *Genre: Drama*},

16 narrow again to {*Products: Movies* AND *Genre: Drama* AND *Director: Spike Lee*}, and

17 then broaden by deselecting a term to {*Products: Movies* AND *Director: Spike Lee*}.

18   In another aspect of the present invention, the user may broaden the navigation

19 state by disjunctively selecting additional terms. For example, a user could start at

20 {*Products: DVDs*}, and then broaden by disjunctively selecting a term to {*Products:*

21 *DVDs* OR *Products: Videos*}, and then narrow by conjunctively selecting a term to

22 {(*Products: DVDs* OR *Products: Videos*) AND *Director: Spike Lee*}.

1   In another aspect of the present invention, the user may narrow the navigation

2   state by negationally selecting additional terms. For example, a user could start at

3   {*Products: DVDs*}, narrow by conjunctively selecting a term to {*Products: DVDs* AND

4   *Genre: Comedy*}, and then narrow by negationally selecting a term to {*Products: DVDs*

5   AND *Genre: Comedy* AND (NOT *Director: Woody Allen*)}.

6   In another aspect of the present invention, the user interface allows users to use

7   free-text search to find terms of interest. In another aspect of the present invention, the

8   user interface also allows users to use free-text search on descriptive information

9   associated with the materials.

10   In another aspect of the present invention, the user interface presents users with

11   context-dependent navigation options for modifying the navigation state. The user

12   interface does not present the user with options whose selection would correspond to no

13   documents in the resulting navigation state. Also, the user interface presents new

14   navigation options as they become relevant. The knowledge base may contain rules that

15   determine when particular attributes or terms are made available to users for navigation.

16   In another aspect of the invention—for example, when the materials correspond to

17   products available for purchase from various sources—the knowledge base includes a

18   catalog of canonical representations that have been aggregated from the materials.

19   In another aspect of the invention, the knowledge base may include definitions of

20   stores, sets of materials that are grouped to be searchable at one time. A store may

21   include documents from one or more domains. An item may be assigned to more than

1   one store. The knowledge base may also include rules to customize navigation for

2   particular stores.

3   In another aspect of the invention, the knowledge base is developed through a

4   multi-stage, iterative process. Workflow management allocates resources to maximize

5   the efficiency of generating and of maintaining the knowledge base. The knowledge base

6   is used to generate data structures that support navigation through a collection of

7   materials. In one aspect of the invention, the navigation system consists of a hierarchy

8   (i.e., a partial order) of navigation states that map expressions of terms to the sets of

9   materials with which those terms are associated. In another aspect of the invention, the

10   navigation states are related by transitions corresponding to terms used to narrow or

11   broaden from one navigation state to another. The navigation states may be fully or

12   partially precomputed, or may be entirely computed at run-time. In another aspect of the

13   invention, implementations of the invention may be scalable through parallel or

14   distributed computation. In addition, implementations of the invention may employ

15   master and slave servers arranged in a hierarchical configuration.

16   4. Brief Description of the Drawings

17   The invention, including these and other features thereof, may be more fully

18   understood from the following description and accompanying drawings, in which:

19   Figure 1 is a view of a user interface to a navigation system in accordance with an

20   embodiment of the present invention.

21   Figure 2 is a view of the user interface of Figure 1, showing a drop-down pick list

22   of navigable terms.

1   Figure 3 is a view of the user interface of Figure 1, showing a navigation state.

2   Figure 4 is a view of the user interface of Figure 1, showing a navigation state.

3   Figure 5 is a view of the user interface of Figure 1, showing a navigation state.

4   Figure 6 is a view of the user interface of Figure 1, showing a navigation state.

5   Figure 7 is a view of the user interface of Figure 1, showing a navigation state.

6   Figure 8 is a view of the user interface of Figure 1, showing a navigation state.

7   Figure 9 is a view of the user interface of Figure 1, showing the result of a free-

8   text search for terms.

9   Figure 10 is a view of the user interface of Figure 1, showing information about a

10   particular document.

11   Figures 11A-C are representative examples of how the range of values for an

12   attribute could be partially ordered in accordance with an embodiment of the present

13   invention.

14   Figure 12 is a block diagram of a process for collecting and classifying documents

15   in accordance with an embodiment of the present invention.

16   Figure 13 is a table illustrating how a set of documents may be classified in

17   accordance with an embodiment of the present invention.

18   Figure 14 is a representative partial order of navigation states in accordance with

19   an embodiment of the present invention.

20   Figure 15 is a block diagram of a process for precomputing a navigation state in

21   accordance with an embodiment of the present invention.

1   Figure 16 is a view of a user interface to a navigation system in accordance with

2   an embodiment of the invention, showing disjunctive selection.

3   Figure 17 is a view of a user interface to a navigation system in accordance with

4   an embodiment of the invention, showing disjunctive selection.

5   Figure 18 is a view of a user interface to a navigation system in accordance with

6   an embodiment of the invention, showing negational selection.

7   Figure 19 is a view of a user interface to a navigation system in accordance with

8   an embodiment of the invention, showing negational selection.

9   Figure 20 is a block diagram of a system and a method for processing a request

10   across multiple servers in accordance with an embodiment of the present invention.

11   Figure 21 is a flow diagram of steps for combining refinement options from slave

12   servers in accordance with an embodiment of the present invention.

13   5. Detailed Description of the Preferred Embodiments

14   **User Interface**

15   In accordance with one embodiment of the present invention, Figure 1 shows a

16   user interface 10 to a hierarchical, data-driven navigation system. The navigation system

17   operates on a collection of documents defined in a knowledge base. As is shown, the

18   user is preferably presented with at least two alternative methods of using the navigation

19   system: (1) by selecting terms to navigate through the collection of documents, or (2) by

20   entering a desired keyword in a search box.

21   The navigation system preferably organizes documents by domain. In accordance

22   with one embodiment of the present invention, the user interface 10 shown in Figures 1-

1   10 is operating on a set of documents that are part of a wine domain. Preferably, a

2   domain defines a portion of the collection of documents that reflects a natural grouping.

3   Generally, the set of attributes used to classify documents in a domain will be a

4   manageable subset of the attributes used to classify the entire collection of documents. A

5   domain definition may be a type of product, e.g., wines or consumer electronics. A

6   domain may be divided into subdomains to further organize the collection of documents.

7   For example, there can be a consumer electronics domain that is divided into the

8   subdomains of televisions, stereo equipment, etc. Documents may correspond to goods

9   or services.

10   The user interface may allow users to navigate in one domain at a time.

11   Alternatively, the user interface may allow the simultaneous navigation of multiple

12   domains, particularly when certain attributes are common to multiple domains.

13   The user interface allows the user to navigate through a collection of navigation

14   states. Each state is composed of an expression of terms and of the set of documents

15   associated with those terms in accordance with that expression. In the embodiment

16   shown in Figures 1-10, users navigate through the collection of navigation states by

17   conjunctively selecting and deselecting terms to obtain the navigation state corresponding

18   to each expression of conjunctively selected terms. Preferably, as in Figure 4, the user

19   interface 10 presents a navigation state by displaying both the list 50 of terms 52 and a

20   list 41 of some or all of the documents 42 that correspond to that state. Preferably, the

21   user interface presents the terms 52 of the navigation state organized by attribute.

1 Preferably, the initial navigation state is a root state that corresponds to no term selections

2 and, therefore, to all of the documents in the collection.

3   As shown in Figure 2, the user interface 10 allows users to narrow the navigation

4 state by choosing a value 28 for an attribute 22, or by replacing the currently selected

5 value with a more specific one (if appropriate). Preferably, the user interface 10 presents

6 users with the options available to narrow the present navigation state, preferably with

7 relevant terms organized by attribute. In some embodiments of the present invention, as

8 shown in Figure 2, users can select values 28 from drop-down lists 26 denoted by

9 indicators 24, that are organized by attributes 22 in the current navigation state. The user

10 interface may present these navigation options in a variety of formats. For example,

11 values can be presented as pictures or as symbols rather than as text. The interface may

12 allow for any method of selecting terms, e.g., mouse clicks, keyboard strokes, or voice

13 commands. The interface may be provided through various media and devices, such as

14 television or WWW, and telephonic or wireless devices. Although discussed herein

15 primarily as a visual interface, the interface may also include an audio component or be

16 primarily audio-based.

17   Preferably, in the present navigation state, the user interface only presents options

18 for narrowing the navigation state that lead to a navigation state with at least one

19 document. This preferred criteria for providing navigation options ensures that there are

20 no "dead ends," or navigation states that correspond to an empty result set.

21   Preferably, the user interface only presents options for narrowing the navigation

22 state if they lead to a navigation state with strictly fewer documents than the present one.

1 Doing so ensures that the user interface does not present the user with choices that are

2 already implied by terms in the current navigation state.

3   Preferably, the user interface presents a new navigation state as soon as the user

4 has chosen a term 28 to narrow the current navigation state, without any further

5 triggering action by the user. Because the system responds to each user with immediate

6 feedback, the user need not formulate a comprehensive query and then submit the query.

7   In accordance with one embodiment of the present invention, as shown in

8 Figures 3 and 4, the user interface 10 may enable broadening of the current navigation

9 state by allowing the user to remove terms 52 from the list 50 of terms conjunctively

10 selected. For example, the interface 10 may provide a list 50 with checkboxes 54 for

11 removing selections and a button 56 to trigger the new search. In the illustrated

12 embodiment, the user can remove conjunctively selected terms 52 in any order and can

13 remove more than one selection 52 at a time.

14   Preferably, the navigation options presented to the user are context-dependent.

15 For example, terms that refine previously selected terms may become navigation options

16 in the resulting navigation state. For example, referring to Figure 5, after the term

17 *Flavors: Wood and Nut Flavors* 52 is conjunctively selected (the user has selected the

18 value Wood and Nut Flavors 23 for the attribute Flavors), Wood and Nut Flavors 23 then

19 appears in the interface for the new navigation state in the list 20 of attributes and allows

20 conjunctive selection of values 28 that relate to that specific attribute for further

21 refinement of the query. The user interface may also present certain attributes that were

22 not presented initially, as they become newly relevant. For example, comparing Figure 3

1   to Figure 2, the attribute French Vineyards 25 appears in the list 20 of attributes only

2   after the user has already conjunctively selected the term *Regions: French Regions* in a

3   previous navigation state. Attributes may be embedded in this way to as many levels as

4   are desired. Presenting attributes as navigation options when those attributes become

5   relevant avoids overwhelming the user with navigation options before those options are

6   meaningful.

7   Additionally, for some attributes 22, multiple incomparable (non-refining)

8   conjunctive selections of values 28 may be applicable. For example, for the attribute

9   Flavor, the values Fruity and Nutty, neither of which refines the other, may both be

10   conjunctively selected so that the terms *Flavors: Fruity* and *Flavors: Nutty* narrow the

11   navigation state. Thus, users may sometimes be able to refine a query by conjunctively

12   selecting multiple values under a single attribute.

13   Preferably, certain attributes will be eliminated as navigation options if they are

14   no longer valid or helpful choices. For example, if all of the documents in the result set

15   share a common term (in addition to the term(s) selected to reach the navigation state),

16   then conjunctive selection of that term will not further refine the result set; thus, the

17   attribute associated with that term is eliminated as a navigation option. For example,

18   comparing Figure 6 with Figure 4, the attribute Wine Types 27 has been eliminated as a

19   navigation option because all of the documents 42 in the result set share the same term,

20   *Wine Types: Appellational Wines*. In preferred embodiments, an additional feature of the

21   interface 10 is that this information is presented to the user as a common characteristic of

22   the documents 42 in the result set. For example, referring to Figure 6, the interface 10

1   includes a display 60 that indicates the common characteristics of the documents 42 in

2   the result set. Removing a term as a navigation option when all of the documents in the

3   result set share that term prevents the user from wasting time by conjunctively selecting

4   terms that do not refine the result set.

5   Preferably, the user interface also eliminates values as navigation options if their

6   selection would result in no documents in the result set. For example, comparing Figure

7   8 to Figure 7, after the user selects the term *Wine Spectator Range: 95 - 100*, the user

8   interface eliminates as navigation options all the values 28, 29 in the list 26 of values for

9   the attribute Appellations 22 except for the values Alexander Valley 29 and Napa Valley

10 29. Alexander Valley 29 and Napa Valley 29 are the only two values in the list 26 of

11 values for the attribute Appellations that return at least one document in the result set; all

12 other values 28 return the empty set. Removing values as navigation options that would

13 result in an empty result set saves the user time by preventing the user from reaching

14 dead-ends.

15 Preferably, the user interface allows users to search for desired words using free-

16 text search. In accordance with one embodiment of the present invention, illustrated in

17 Figure 9, a search box 30 preferably allows users to perform a free-text search for terms

18 of interest, rather than performing a full-text search of the documents themselves.

19 Preferably, the user interface responds to such a search by presenting a list 32 of terms 34

20 including terms organized by attribute 36, and allowing the user to select from among

21 them. Preferably, the user interface responds to the user's selection by presenting the

22 user with the navigation state corresponding to the selection of that term. The user may

1   then either navigate from that state (i.e., by narrowing or broadening it) or perform

2   additional free-text searches for terms.

3   Preferably, the user interface 10 presents a full or partial list 41 of the documents

4   that correspond to the current navigation state. Preferably, if a user is interested in a

5   particular document 42, the user may select it and obtain a record 70 containing further

6   information about it, including the list 72 of terms 74 that are associated with that

7   document, as shown in Figure 10. Preferably, the user interface 10 allows the user to

8   conjunctively select any subset of those terms 74 and thereby navigate to the navigation

9   state that corresponds to the selected term expression.

10   Preferably, the user interface 10 also offers navigation options that directly link to

11   an associated navigation state that is relevant to, but not necessarily a generalization or

12   refinement of, the present navigation state. These links preferably infer the user's

13   interests from the present navigation state and enable the user to cross-over to a related

14   topic. For example, if the user is visiting a particular navigation state in a food domain,

15   links may direct the user to navigation states of wines that would complement those foods

16   in the wine domain.

17   In accordance with another embodiment of the present invention, the user is

18   preferably presented with additional methods of using the navigation system such as: (1)

19   by conjunctively selecting terms, (2) by disjunctively selecting terms, (3) by negationally

20   selecting terms, or (4) by entering a desired keyword in a search box.

21   In another aspect of the present invention, the user may broaden the navigation

22   state by disjunctively selecting additional terms. For example, a user could start at

1 {*Products: DVDs*}, and then broaden by disjunctively selecting a term to {*Products:*

2 *DVDs* OR *Products: Videos*}, and then narrow by conjunctively selecting a term to

3 {*(Products: DVDs* OR *Products: Videos)* AND *Director: Spike Lee*}. Figure 16 shows a

4 user interface 300 to a hierarchical, data-driven navigation system. The user interface

5 300 is operating on a collection of records relating to mutual funds. The interface 300

6 presents navigation options, including a list of attributes 310 relating to mutual funds and

7 a list of terms 314 for a particular attribute 312, such as Fund Family, under consideration

8 by a user. A selected term 316 is highlighted. As shown, the attribute-value pair {*Fund*

9 *Family: Fidelity Investments*} has previously been selected. The illustrated navigation

10 system allows the user to select attribute-value pairs disjunctively. As shown in Figure

11 17, after the user subsequently selects {*Fund Family: Vanguard Group*}in addition, the

12 interface 300 presents a new navigation state {*Fund Family: Fidelity Investments* OR

13 *Fund Family: Vanguard Group*}, including mutual funds 320 that match either selected

14 attribute-value pair. Accordingly, both selected attribute-value pairs 316 are highlighted.

15 In some embodiments, for example, to reduce computational requirements, disjunctive

16 combination of attribute-value pairs may be limited to mutually incomparable attribute-

17 value pairs that correspond to the same attribute.

18   In another aspect of the present invention, the user may narrow the navigation

19 state by negationally selecting additional terms. For example, a user could start at

20 {*Products: DVDs*}, narrow by conjunctively selecting a term to {*Products: DVDs* AND

21 *Genre: Comedy*}, and then narrow by negationally selecting a term to {*Products: DVDs*

22 AND *Genre: Comedy* AND (NOT *Director: Woody Allen)*}. Figure 18 shows another

1   interface 400 to a hierarchical, data-driven navigation system. The user interface 400 is

2   operating on a collection of records relating to entertainment products. The user interface

3   400 includes a header 410 and a navigation area 412. The header 410 indicates the

4   present navigation state {*Products: DVDs* AND *Genre:Drama*}, and implies the

5   refinement options currently under consideration by the user. The leader "Not Directed

6   By" 414 indicates a negational operation with respect to the Director attribute. The

7   interface lists the attribute-value pairs 416 that can be combined with the expression for

8   the present navigation state under this operation. As shown in Figure 19, after the user

9   selects the term *Director: Martin Scorsese*, the interface 400 presents a new navigation

10   state {*Products: DVDs* AND *Genre:Drama* AND (NOT *Director: Martin Scorsese*}.

11   Although the interface to the navigation system has been described herein as a

12   user interface, the interface could provide other forms of access to the navigation system.

13   In alternative embodiments, the interface may be an applications program interface to

14   allow access to the navigation system for or through other applications. The interface

15   may also enhance the functionality of an independent data-oriented application. The

16   interface may also be used in the context of a WWW-based application or an XML-based

17   application. The navigation system may also support multiple interface modes

18   simultaneously. The navigation system may be made available in a variety of ways, for

19   example via wireless communications or on handheld devices.

20   **Knowledge Base**

21   Preferably, the navigation system stores all information relevant to navigation in a

22   knowledge base. The knowledge base is the repository of information from two

1 processes: taxonomy definition and classification. Taxonomy definition is the process of

2 identifying the relevant attributes to characterize documents, determining the acceptable

3 values for those attributes (such as a list or range of values), and defining a partial order

4 of refinement relationships among terms (attribute-value pairs). Classification is the

5 process of associating terms with documents. The knowledge base may also be used to

6 maintain any information assets that support these two processes, such as domains,

7 classification rules and default expectations. Additionally, the knowledge base may be

8 used to maintain supplementary information and materials that affect users' navigation

9 experience.

10 The taxonomy definition process identifies a set of attributes that appropriately

11 characterize documents. A typical way to organize the taxonomy definition process is to

12 arrange the collections of documents into domains, which are sets of documents that

13 conform to a natural grouping and for which a manageable number of attributes suffice to

14 effectively distinguish and navigate among the documents in that domain. The knowledge

15 base preferably includes a characterization of each domain, which might include rules or

16 default expectations concerning the classification of documents in that domain.

17 The taxonomy definition process also identifies a full set of values, at varying

18 levels of specificity when appropriate, for each attribute. The values preferably identify

19 the specific properties of the documents in the collection. The values may be enumerated

20 explicitly or defined implicitly. For example, for a "color" attribute, a full set of valid

21 color values may be specified, but for a "price" or "date" attribute, a range within which

22 the values may fall or a general data type, without defining a range, may be specified.

1   The process of identifying these values may include researching the domain or analyzing

2   the collection of documents.

3   The taxonomy definition process also defines a partial order of refinement

4   relationships among terms (attribute-value pairs). For example, the term *Origin: France*

5   could refine the term *Origin: Europe*. The refinement relationship is transitive and

6   antisymmetric but not necessarily total. Transitivity means that, if term A refines term B

7   and term B refines term C, then term A refines term C. For example, if *Origin: Paris*

8   refines *Origin: France* and *Origin: France* refines *Origin: Europe*, then *Origin: Paris*

9   refines *Origin: Europe*. Antisymmetry means that, if two terms are distinct, then both

10   terms cannot refine each other. For example, if *Origin: Paris* refines *Origin: France*,

11   then *Origin: France* does not refine *Origin: Paris*.

12   Further, the partial order of refinement relationships among terms is not

13   necessarily a total one. For example, there could be two terms, *Origin: France* and

14   *Origin: Spain*, such that neither term refines the other. Two terms with this property are

15   said to be incomparable. Generally, a set of two or more terms is mutually incomparable

16   if, for every pair of distinct terms chosen from that set, the two terms are incomparable.

17   Typically, but not necessarily, two terms with distinct attributes will be incomparable.

18   Given a set of terms, a term is a maximal term in that set if it does not refine any

19   other terms in the set, and it is a minimal term in that set if no other term in the set refines

20   it. For example, in the set {*Origin: France, Origin: Paris, Origin: Spain, Origin:*

21   *Madrid*}, *Origin: France* and *Origin: Spain* are maximal, while *Origin: Paris* and

1   *Origin: Madrid* are minimal. In the knowledge base, a term is a root term if it does not

2   refine any other terms and a term is a leaf term if no other term refines it.

3   Figures 11A, 11B, and 11C illustrate attributes 112 and values 114, arranged in

4   accordance with the partial order relationships, that could be used for classifying wines.

5   The attributes 112 are Type/Varietal, Origin, and Vintage. Each attribute 112

6   corresponds to a maximal term for that attribute. An attribute 112 can have a flat set of

7   mutually incomparable values (e.g., Vintage), a tree of values (e.g., Origin), or a general

8   partial order that allows a value to refine a set of two or more mutually incomparable

9   values (e.g., Type/Varietal). The arrows 113 indicate the refinement relationships among

10 values 114.

11 Attributes and values may be identified and developed in several ways, including

12 manual or automatic processing and the analysis of documents. Moreover, this kind of

13 analysis may be top-down or bottom-up; that is, starting from root terms and working

14 towards leaf terms, or starting from leaf terms and working towards root terms. Retailers,

15 or others who have an interest in using the present invention to disseminate information,

16 may also define attributes and terms.

17 The classification process locates documents in the collection of navigation states

18 by associating each document with a set of terms. Each document is associated with a set

19 of mutually incomparable terms, e.g., {*Type/Varietal: Chianti, Origin: Italy, Vintage:*

20 *1996*}, as well as any other desired descriptive information. If a document is associated

21 with a given term, then the document is also associated with all of the terms that the

22 given term refines.

1   The classification process may proceed according to a variety of workflows.

2   Documents may be classified in series or in parallel, and the automatic and manual

3   classification steps may be performed one or more times and in any order. To improve

4   accuracy and throughput, human experts may be assigned as specialists to oversee the

5   classification task for particular subsets of the documents, or even particular attributes for

6   particular subsets of the documents. In addition, the classification and taxonomy

7   processes may be interleaved, especially as knowledge gained from one process allows

8   improvements in the other.

9   Figure 12 illustrates the stages in a possible flow for the classification process

10   250. The data acquisition step 252, that is, the collection of documents for the database,

11   may occur in several different ways. For example, a retailer with a product catalog over

12   which the navigation system will operate might provide a set of documents describing its

13   products as a pre-defined set. Alternatively, documents may be collected from one

14   source, e.g., one Web site, or from a number of sources, e.g., multiple Web sites, and then

15   aggregated. If the desired documents are Web pages, the documents may be collected by

16   appropriately crawling the Web, selecting documents, and discarding documents that do

17   not fit in the domain. In the data translation step 254, the collected documents are

18   formatted and parsed to facilitate further processing. In the automatic classification step

19   256, the formatted and parsed documents are processed in order to automatically

20   associate documents with terms. In the manual classification step 258, human reviewers

21   may verify and amend the automatic classifications, thereby ensuring quality control.

22   Preferably, any rules or expectations violated in either the automatic classification step

1   256 or the manual classification step 258 would be flagged and presented to human

2   reviewers as part of the manual classification step 258. If the collection of documents is

3   divided into domains, then there will typically be rules that specify a certain minimal or

4   preferred set of attributes used to classify documents from each domain, as well as other

5   domain-specific classification rules. When the classification process is complete, each

6   document will have a set of terms associated with it, which locate the document in the

7   collection of navigation states.

8   In Figure 13, table 180 shows a possible representation of a collection of

9   classified wine bottles. Preferably, each entry is associated with a document number 182,

10   which could be a universal identifier, a name 184, and the associated terms 186. The

11   name is preferably descriptive information that could allow the collection to be accessed

12   via a free-text search engine as well as via the term-based navigation system.

13   In another aspect of the invention, the knowledge base also includes a catalog of

14   canonical representations of documents. Each catalog entry represents a conceptually

15   distinct item that may be associated with one or more documents. The catalog allows

16   aggregation of profile information from multiple documents that relate to the item,

17   possibly from multiple sources. For example, if the same wine is sold by two vendors,

18   and if one vendor provides vintage and geographic location information and another

19   provides taste information, that information from the two vendors can be combined in the

20   catalog entry for that type of wine. The catalog may also improve the efficiency of the

21   classification process by eliminating duplicative profiling. In Figure 12, the catalog

22   creation step 260 associates classified documents with catalog entries, creating new

1   catalog entries when appropriate. For ease of reference, an item may be uniquely

2   identified in the catalog by a universal identifier.

3   The knowledge base may also define stores, where a store is a subcollection of

4   documents that are grouped to be searchable at one time. For example, a particular

5   online wine merchant may not wish to display documents corresponding to products sold

6   by that merchant's competitors, even though the knowledge base may contain such

7   documents. In this case, the knowledge base can define a store of documents that does

8   not include wines sold by the merchant's competitors. In Figure 12, the store creation

9   step 262 may define stores based on attributes, terms, or any other properties of

10   documents. A document may be identified with more than one store. The knowledge

11   base may also contain attributes or terms that have been customized for particular stores.

12   In Figure 12, the export process step 264 exports information from the knowledge

13   base to another stage in the system that performs further processing necessary to generate

14   a navigable data structure.

15   **Navigation States**

16   The navigation system represents, explicitly or implicitly, a collection of

17   navigation states. A navigation state can be represented either by an expression of terms,

18   or by the subset of the collection of documents that correspond to the term expression.

19   By way of example, types of navigation states include conjunctive navigation

20   states, disjunctive navigation states and negational navigation states. Conjunctive

21   navigation states are a special case of navigation states in which the term expression is

22   conjunctive—that is, the expression combines terms using only the AND operator.

1   Conjunctive navigation states are related by a partial order of refinement that is derived

2   from the partial order that relates the terms.

3   In one aspect of the present invention, a conjunctive navigation state has two

4   representations. First, a conjunctive navigation state corresponds to a subset of the

5   collection of documents. Second, a conjunctive navigation state corresponds to a

6   conjunctive expression of mutually incomparable terms. Figure 14 illustrates some

7   navigation states for the documents and terms based on the wine example discussed

8   above. For example, one navigation state 224 is {*Origin: South America*} (documents

9   #1, #4, #5); a second navigation state 224 is {*Type/Varietal: White* AND *Origin: United*

10   *States*} (documents #2, #9). The subset of documents corresponding to a conjunctive

11   navigation state includes the documents that are commonly associated with all of the

12   terms in the corresponding expression of mutually incomparable terms. At the same

13   time, the expression of mutually incomparable terms corresponding to a conjunctive

14   navigation state includes all of the minimal terms from the terms that are common to the

15   subset of documents, i.e., the terms that are commonly associated with every document in

16   the subset. A conjunctive navigation state is preferably unique and fully specified; for a

17   particular conjunctive expression of terms, or for a given set of documents, there is no

18   more than one corresponding conjunctive navigation state.

19   One way preferred to define the collection of conjunctive navigation states is to

20   uniquely identify each conjunctive navigation state by a canonical conjunctive expression

21   of mutually incomparable terms. A two-step mapping process that maps an arbitrary

22   conjunctive expression of terms to a canonical conjunctive expression of mutually

1   incomparable terms creates states that satisfy this property. In the first step of the

2   process, an arbitrary conjunctive expression of terms is mapped to the subset of

3   documents that are associated with all of those terms. Recalling that if a document is

4   associated with a given term, then the document is also associated with all of the terms

5   that the given term refines, in the second step of the process, this subset of documents is

6   mapped to the conjunctive expression of minimal terms among the terms that are

7   common to all of the documents in that document set. The result of this second step is a

8   conjunctive expression of mutually incomparable terms that uniquely identifies the

9   corresponding subset of documents, and, hence, is a canonical representation for a

10   conjunctive navigation state. By way of illustration, referring to the wine example in

11   Figure 14, the term expression {*Origin: France*} maps to the subset of documents

12   {documents #8, #11}, which in turn maps to the canonical term expression

13   {*Type/Varietal: Red* AND *Origin: France*}.

14   The conjunctive navigation states 222, 224, 226 are related by a partial order of

15   refinement relationships 220 derived from the partial order that relates terms. This partial

16   order can be expressed in terms of either the subsets of documents or the term

17   expressions that define a conjunctive navigation state. Expressed in terms of subsets of

18   documents, a navigation state A refines a navigation state B if the set of documents that

19   corresponds to state A is a subset of the set of documents that corresponds to state B.

20   Expressed in terms of term expressions, a conjunctive navigation state A refines a

21   conjunctive navigation state B if all of the terms in state B either are in state A or are

22   refined by terms in state A. Referring to Figure 14, the navigation state 226

1   corresponding to the term expression {*Type/Varietal: Red* AND *Origin: Chile*}

2   (document #4) refines the navigation state 224 corresponding to {*Origin: Chile*}

3   (documents #4, #5). Since the refinement relationships among navigation states give rise

4   to a partial order, they are transitive and antisymmetric. In the example, {*Type/Varietal:*

5   *Red* AND *Origin: Chile*} (document #4) refines {*Origin: Chile*} (documents #4, #5) and

6   {*Origin: Chile*} (documents #4, #5) refines {*Origin: South America*} (documents #1, #4,

7   #5); therefore, {*Type/Varietal: Red* AND *Origin: Chile*} (document #4) refines {*Origin:*

8   *South America*} (documents #1, #4, #5). The root navigation state 222 is defined to be

9   the navigation state corresponding to the entire collection of documents. The leaf

10   navigation states 226 are defined to be those that cannot be further refined, and often

11   (though not necessarily) correspond to individual documents. There can be arbitrarily

12   many intermediate navigation states 224 between the root 222 and the leaves 226. Given

13   a pair of navigation states A and B where B refines A, there can be multiple paths of

14   intermediate navigation states 224 connecting A to B in the partial order. For

15   convenience of definition in reference to the implementation described herein, a

16   navigation state is considered to refine itself.

17   A user browses the collection of documents by visiting a sequence of one or more

18   navigation states typically starting at the root navigation state 222. In one embodiment of

19   the present invention, there are three basic modes of navigation among these states. The

20   first mode is refinement, or moving from the current navigation state to a navigation state

21   that refines it. The user can perform refinement either by adding a term through

22   conjunctive selection to the current navigation state or by refining a term in the current

1   navigation state; i.e., replacing a term with a refinement of that term. After the user adds

2   or refines a term, the new term expression can be mapped to a canonical term expression

3   according to the two-step mapping described above. The second mode is generalization,

4   or moving from the current navigation state to a more general navigation state that the

5   current state refines. The user can perform generalization either by removing a term from

6   the current navigation state or by generalizing a term in the current navigation state; i.e.,

7   replacing a current term with a term that the current term refines. After the user removes

8   or generalizes a term, the new term expression can be mapped to a canonical term

9   expression. The third mode is simply creating a query in the form of a desired term

10   expression, which again can be mapped to a canonical term expression to obtain a

11   navigation state.

12   In other embodiments of the present invention, there are additional modes of

13   navigation. In systems that support the corresponding types of navigation states, these

14   modes may include generalization of the navigation state through disjunctive selection, as

15   shown in Figure 16, as well as refinement of the navigation state through negational

16   selection, as shown in Figure 17. In general, terms can be combined using Boolean logic.

17   Although term expressions that are not conjunctive do not necessarily have canonical

18   forms, some implementations may be based on a system that uses a collection of

19   conjunctive navigation states. One implementation is based on logical equivalence rules

20   as described below.

21   **Implementation**

1   The knowledge base is transferred to a navigable data structure in order to

2   implement the present invention. The navigation states may be fully precomputed,

3   computed dynamically at run-time, or partially precomputed. A cache may be used to

4   avoid redundant computation of navigation states.

5   In preferred embodiments, the collection of conjunctive navigation states may be

6   represented as a graph—preferably, a directed acyclic multigraph with labeled edges. A

7   graph is a combinatorial structure consisting of nodes and edges, where each edge links a

8   pair of nodes. The two nodes linked by an edge are called its endpoints. With respect to

9   the present invention, the nodes correspond to conjunctive navigation states, and the

10   edges represent transitions that refine from one conjunctive navigation state to another.

11   Since refinement is directional, each edge is directed from the more general node to the

12   node that refines it. Because there is a partial order on the navigation states, there can be

13   no directed cycles in the graph, i.e., the graph is acyclic. Preferably, the graph is a

14   multigraph, since it allows the possibility of multiple edges connecting a given pair of

15   nodes. Each edge is labeled with a term. Each edge has the property that starting with

16   the term set of the more general end point, adding the edge term, and using the two-step

17   map to put this term set into canonical form leads to a refinement which results in the

18   navigation state that is the other endpoint. That is, each edge represents a refinement

19   transition between nodes based on the addition of a single term.

20   The following definitions are useful for understanding the structure of the graph:

21   descendant, ancestor, least common ancestor (LCA), proper ancestor, proper descendant,

22   and greatest lower bound (GLB). These definitions apply to the refinement partial order

1   among terms and among nodes. If A and B are terms and B refines A, then B is said to

2   be a descendant of A and A is said to be an ancestor of B. If, furthermore, A and B are

3   distinct terms, then B is said to be a proper descendant of A and A is said to be a proper

4   ancestor of B. The same definitions apply if A and B are both nodes.

5   If C is an ancestor of A and C is also an ancestor of B, then C is said to be a

6   common ancestor of A and B, where A, B, and C are either all terms or all nodes. The

7   minimal elements of the set of common ancestors of A and B are called the least common

8   ancestors (LCAs) of A and B. If no term has a pair of incomparable ancestors, then the

9   LCA of two terms—or of two nodes—is unique. For example, the LCA of *Origin:*

10   *Argentina* and *Origin: Chile* is *Origin: South America* in the partial order of terms 110 of

11   Figure 11B. In general, however, there may be a set of LCAs for a given pair of terms or

12   nodes.

13   In an implementation that fully precomputes the collection of nodes, computation

14   of the nodes in the graphs is preferably performed bottom-up.

15   The leaf nodes in the graph—that is, the nodes corresponding to leaf navigation

16   states—may be computed directly from the classified documents. Typically, but not

17   necessarily, a leaf node will correspond to a set containing a single document. The

18   remaining, non-leaf nodes are obtained by computing the LCA-closure of the leaf

19   nodes—that is, all of the nodes that are the LCAs of subsets of the leaf nodes.

20   The edges of the graph are determined according to a refinement function, called

21   the R function for notational convenience. The R function takes as arguments two nodes

22   A and B, where A is a proper ancestor of B, and returns the set of maximal terms such

1   that, if term C is in R (A, B), then refining node A with term C results in a node that is a

2   proper descendant of A and an ancestor (not necessarily proper) of B. For example, in

3   Figure 14, R ({*Type/Varietal: Red*}, {*Type/Varietal: Merlot* AND *Origin: Argentina*

4   AND *Vintage: 1998*}) = {*Type/Varietal: Merlot* AND *Origin: South America* AND

5   *Vintage: 1998*}. If B_1 is an ancestor of B_2, then R (A, B_1) is a subset of R (A, B_2)—

6   assuming that A is a proper ancestor of both B_1 and B_2. For example, R ({*Type/Varietal:*

7   *Red*}, {*Type/Varietal: Red* AND *Origin: South America*}) = {*Origin: South America*}.

8   In the graph, the edges between nodes A and B will correspond to a subset of the

9   terms in R (A, B). Also, no two edges from a single ancestor node A use the same term

10   for refinement. If node A has a collection of descendant nodes {B_1, B_2,...} such that

11   term C is in all of the R (A, B_i), then the only edge from node A with term C goes to

12   LCA (B_1, B_2,...), which is guaranteed to be the unique maximal node among the B_i. In

13   Figure 14, for example, the edge from node {*Type/Varietal: Red*} with term *Origin:*

14   *South America* goes to node {*Type/Varietal: Red* AND *Origin: South America*} rather

15   than to that node's proper descendants {*Type/Varietal: Merlot* AND *Origin: South*

16   *America* AND *Vintage: 1998*} and {*Type/Varietal: Red* AND *Origin: Chile*}. The LCA-

17   closure property of the graph ensures the existence of a unique maximal node among the

18   B_i. Thus, each edge maps a node-term pair uniquely to a proper descendant of that node.

19   The LCA-closure of the graph results in the useful property that, for a given term

20   set S, the set of nodes whose term sets refine S has a unique maximal node. This node is

21   called the greatest lower bound (GLB) of S.

1   The graph may be computed explicitly and stored in a combinatorial data

2   structure; it may be represented implicitly in a structure that does not necessarily contain

3   explicit representations of the nodes and edges; or it may be represented using a method

4   that combines these strategies. Because the navigation system will typically operate on a

5   large collection of documents, it is preferred that the graph be represented by a method

6   that is scalable.

7   The graph could be obtained by computing the LCAs of every possible subset of

8   leaf nodes. Such an approach, however, grows exponentially in the number of leaf nodes,

9   and is inherently not scalable. An alternative strategy for obtaining the LCA closure is to

10   repeatedly consider all pairs of nodes in the graph, check if each pair's LCA is in the

11   graph, and add that LCA to the graph as needed. This strategy, though a significant

12   improvement on the previous one, is still relatively not scalable.

13   A more efficient way to precompute the nodes is to process the document set

14   sequentially, compute the node for each document, and add that node to the graph along

15   with any other nodes necessary to maintain LCA-closure. The system stores the nodes

16   and edges as a directed acyclic multigraph. The graph is initialized to contain a single

17   node corresponding to the empty term set, the root node. Referring to Figure 15, in

18   process 230 for inserting a new node into the graph, in step 232, for each new document

19   to be inserted into the graph that does not correspond to an existing node, the system

20   creates a new node. In step 234, before inserting the new node into the graph, the system

21   recursively generates and inserts any missing LCA nodes between the root node (or

22   ancestor node) and the new node. To ensure LCA-closure after every node insertion, the

1   system inserts the document node last, in steps 236 and 238, after inserting all the other

2   nodes that are proper ancestors of it.

3   Inserting a new node requires the addition of the appropriate edges from ancestors

4   to the node, in step 236, and to descendants out of the new node, in step 238. The edges

5   into the node are preferably determined by identifying the ancestors that have refinement

6   terms that lead into the new node and do not already have those refinement terms used on

7   edges leading to intermediate ancestors of the new node. The edges out of the node are

8   preferably determined by computing the GLB of the new node and appropriately adding

9   edges from the new node to the GLB and to nodes to which the GLB has edges.

10   The entire graph of conjunctive navigation states may be precomputed by

11   following the above procedures for each document in the collection. Computation of

12   other types of navigation states is discussed below. Precomputing of the graph may be

13   preferred where the size of the graph is manageable, or if users are likely to visit every

14   navigation state with equal probability. In practice, however, users typically visit some

15   navigation states more frequently than others. Indeed, as the graph gets larger, some

16   navigation states may never be visited at all. Unfortunately, reliable predictions of the

17   frequency with which navigation states will be visited are difficult. In addition, it is

18   generally not practical to precompute the collection of navigation states that are not

19   conjunctive, as this collection is usually much larger than the collection of conjunctive

20   navigation states.

21   An alternative strategy to precomputing the navigation states is to create indexes

22   that allow the navigation states to be computed dynamically. Specifically, each

1	document can be indexed by all of the terms that are associated with that document or

2	that have refinements associated with that document. The resulting index is generally

3	much smaller in size than a data structure that stores the graph of navigation states. This

4	dynamic approach may save space and precomputation time, but it may do so at the cost

5	of higher response times or greater computational requirements for operation. A dynamic

6	implementation may use a one-argument version of the R function that returns all

7	refinement terms from a given navigation state, as well a procedure for computing the

8	GLB of a term set.

9	It is also possible to precompute a subset of the navigation states. It is preferable

10	to precompute the states that will cost the most to compute dynamically. For example, if

11	a state corresponds to a large subset of the documents, it may be preferable to compute it

12	in advance. In one possible partial precomputation approach, all navigation states,

13	particularly conjunctive ones, corresponding to a subset of documents above a threshold

14	size may be precomputed. Precomputing a state is also preferable if the state will be

15	visited frequently. In some instances it may be possible to predict the frequency with

16	which a navigation state will be visited. Even if the frequency with which a navigation

17	state will be visited cannot be predicted in advance, the need to continually recompute

18	can be reduced by caching the results of dynamic computation. Most recently or most

19	frequently visited states may be cached.

20	As described above with respect to the interface, the system supports at least three

21	kinds of query operations—namely refinement, generalization, and query by specifying

22	an expression of terms. These operations may be further described in terms of the graph.

1   For query refinement, the system enumerates the terms that are on edges from the node

2   corresponding to the current navigation state. When the user selects a term for

3   refinement, the system responds by presenting the node to which that edge leads.

4   Similarly, for query generalization options, the system enumerates and selects edges that

5   lead to (rather than from) the node corresponding to the current navigation state.

6   Alternatively, query generalization may be implemented as a special case of query by

7   specifying a set of terms. For query by specifying a set of keywords, the system creates a

8   virtual node corresponding to the given term set and determines the GLB of the virtual

9   node in the graph. If no GLB is found, then there are no documents that satisfy the

10   query. Otherwise, the GLB node will be the most general node in the graph that

11   corresponds to a navigation state where all documents satisfy the query.

12   The above discussion focuses on how the system represents and computes

13   conjunctive navigation states. In some embodiments of the present invention, the user

14   interface only allows users to navigate among the collection of conjunctive navigation

15   states. In other embodiments, however, users can navigate to navigation states that are

16   not conjunctive. In particular, when the system supports navigation states that are not

17   conjunctive, the user interface may allow users to select terms disjunctively or

18   negationally.

19   If the system includes navigation states that are both conjunctive and disjunctive

20   (e.g., {(*Products: DVDs* OR *Products: Videos*) AND *Director: Spike Lee*}), then in some

21   embodiments, the system only precomputes a subset of the states, particularly if the total

22   number of navigation states is likely to be too large to maintain in memory or even

1   secondary (e.g., disk) storage. By using rules for equivalence of Boolean expressions, it

2   is possible to express any navigation state that mixes conjunction and disjunction in terms

3   of a union of conjunctive navigation states. The above example can be rewritten as

4   {(*Products: DVDs* AND *Director: Spike Lee*) OR (*Products: Videos* AND *Director:*

5   *Spike Lee*)}. This approach leads to an implementation combining conjunctive and

6   disjunctive navigation states based on the above discussion, regardless of whether all,

7   some, or none of the graph of conjunctive navigation states is precomputed.

8   In preferred embodiments, disjunctive selections may be made within, but not

9   between, attributes. When determining the set of disjunctive generalizations, the system

10 does not consider other terms from the attribute of the given disjunction to be in the

11 navigation state. For example, if the navigation state is {*Type/Varietal: Red* AND

12 *Origin: Chile*} and the system is allowing the disjunctive selection of other countries of

13 origin, then the GLB and R function will be applied to the set {*Type/Varietal: Red*}

14 rather than to {*Type/Varietal: Red* AND *Origin: Chile*}. Accordingly, the other terms for

15 the attribute of country of origin that are incomparable to "Chile" become generalization

16 options for the navigation state.

17 If the system includes navigation states that use negation (e.g., {*Products: DVDs*

18 AND *Genre: Comedy* AND (NOT *Director: Woody Allen*)}), then the negationally

19 selected terms can be applied to navigation states as a post-process filtering operation.

20 The above example can be implemented by taking the conjunctive navigation state

21 {*Products: DVDs* AND *Genre: Comedy*} and applying a filter to it that excludes all

22 movies associated with the term *Director: Woody Allen*. This approach leads to an

1   implementation including negational navigation states based on the above discussion,

2   regardless of whether all, some, or none of the graph of conjunctive navigation states is

3   precomputed.

4   As with disjunction, when determining the set of negational generalizations, the

5   system does not consider other terms from the attribute of the given negation to be in the

6   navigation state. For example, if the navigation state is {*Medium: Compact Disc* AND

7   *Artist: Prince*} and the system is allowing the negational selection of other artists (e.g.,

8   {*Artist: Prince* AND NOT (*Artist: The Revolution*)}), then the GLB and R function will

9   be applied to the set {*Medium: Compact Disc*} rather than to {*Medium: Compact Disc*

10   AND *Artist: Prince*}.

11   Another aspect of the present invention is its scalability through parallel or

12   distributed computation. One way to define scalability in a navigation system is in terms

13   of four problem dimensions: the number of materials in the collection, the number of

14   terms associated with each material in the collection, the rate at which the system

15   processes queries (throughput), and the time necessary to process a query (latency). In

16   this definition, a system as scalable if it can be scaled along any of these four dimensions

17   at a subquadratic cost. In other words:

18   1. If the number of materials in the collection is denoted by the variable n_1 and the

19   other three problem dimensions are held constant, then the resource requirements

20   are subquadratic in n_1.

1   2. If the number of terms associated with each material in the collection is denoted

2   by the variable n_2 and the other three problem dimensions are held constant, then

3   the resource requirements are subquadratic in n_2.

4   3. If the number of queries that the system processes per second (i.e., the

5   throughput) is denoted by the variable n_3 and the other three problem dimensions

6   are held constant, then the resource requirements are subquadratic in n_3.

7   4. If the time necessary to process a query (i.e., the latency) is denoted by the

8   variable n_4 and the other three problem dimensions are held constant, then the

9   resource requirements are subquadratic in $1/n_4$.

10   Preferably, these resource requirements would be not only subquadratic, but

11   linear. Also included within the concept of scalability, there is an allowance for overhead

12   in creating a network of distributed resources. Typically, this overhead will be

13   logarithmic, since the resources may be arranged in a hierarchical configuration of

14   bounded fan-out.

15   In some embodiments, the present invention surmounts the limitations of a single

16   computational server's limited resources by allowing for distributing the task of

17   computing the information associated with a navigation state onto a hierarchy of multiple

18   computational servers that act in parallel.

19   One insight that drives this aspect of the present invention is that it is possible to

20   partition the collection of materials among multiple "slave" servers, all of which

21   implement the single-server algorithm for multidimensional navigation, and then to have

1   a "master" server compute navigation states by passing requests onto the set of slave

2   machines and combining the responses. From the outside, the collection of servers

3   appears to act like a single server, but with far greater computational resources than

4   would be possible on a single computational device. Indeed, the distinction between

5   master and slave servers is arbitrary; a slave server can itself have slaves, thus creating a

6   nested hierarchy of servers. Such nesting is useful when the number of slaves exceeds

7   the fan-out capability of a single master server. An exemplary embodiment of such a

8   system is illustrated in Figure 20. In the hierarchical arrangement 500, a master server

9   520 works with slave servers 530, 540. In the hierarchical arrangement shown, slave

10   servers 530 are in turn master servers with respects to slave servers 540. The search

11   results are made available to a user on a terminal 510 through a user interface in

12   accordance with the present invention.

13   The collection of materials may be partitioned by splitting (arbitrarily or

14   otherwise) the materials into disjoint subsets, where each subset is assigned to its own

15   server. The subsets may be roughly equal in size, or they might vary in size to reflect the

16   differing computational resources available to each server.

17   The algorithm for distributing the task of computing the information associated

18   with a navigation state includes three steps. The steps of the algorithm are indicated in

19   Figure 20. In the first step, the query, which is a request for a valid navigation state, is

20   submitted to the master server 520, which forwards the query to each of the slave servers

21   530. If the servers are nested, the requests are forwarded through the hierarchy of servers

22   500 until they reach the leaf servers 540 in the hierarchy. In the second step, each slave

1   server 530, 540 processes the query independently, based on the subset of the collection

2   of materials that is in its partition. In the third step, the master server 520 combines the

3   responses from the slave servers to produce a response for the original query. The master

4   server 520 returns the response to the terminal 510.

5   The master server receives the original request and farms it out to the slave

6   servers. Thus, in preferred embodiments, the only computation performed by the master

7   server is to combine the results from the slave servers. Each slave server that receives a

8   request computes the navigation state based on the subset of the collection assigned to it.

9   The computation may involve any combination of conjunction, disjunction, and negation.

10   The master server, in contrast, only performs a combination step. The

11   combination step involves producing a valid navigation state, including documents and

12   corresponding refinement options, from the responses from the slave servers. Since the

13   collection of materials has been partitioned into disjoint subsets, the documents identified

14   by each of the slave servers can be combined efficiently as a disjoint union. Combining

15   the various refinement options returned by each of the slave servers may require

16   additional processing, as described below.

17   The slave servers all process the same query, but on different partitions of the

18   collection of materials. They will generally return different sets of refinement options

19   because a set of refinement options that is valid for one partition may be invalid for

20   another. If the different sets are disjoint, and if the refinement options involve terms that

21   do not themselves have refinement relationships, then the combination is a disjoint union.

1   Typically, there will be some overlap among the different sets of refinement

2   options returned by each slave server. If the sets are not disjoint, duplicates can be

3   eliminated in this combination step.

4   When there are refinement relationships among the terms that are refinement

5   options returned by the slave servers, the combination algorithm computes, for every set

6   of related terms, the least common ancestor or ancestors (LCA) of the terms, as defined

7   by the partial order among the terms. One algorithm for combining the refinement

8   options is outlined in Figure 21. In step 552, the master server receives and takes the

9   union of all of the terms, x_1, x_2, ... x_n, returned as refinement options for the navigation

10   state from the slave servers. In step 554, the master server computes the set of ancestors

11   A_1, A_2, ... An, for each of the terms, x_1, x_2, ... xn, respectively. In step 556, the master

12   server computes the intersection A of all of the sets of ancestors, A_1, A_2, ... An. In step

13   558, the master server computes the set M of minimal terms in A. The set M, formed of

14   the least common ancestors of the terms x_1, x_2, ... x_n, returned by the slave servers, is the

15   set of refinement options corresponding to the result navigation state. This combination

16   procedure is applied whether the refinement options are conjunctive, disjunctive, or

17   negational.

18   In summary, the master server receives a request for a navigation state, forwards

19   this request to each of the slave servers, combines their results with a union operation,

20   and then computes, for every set of terms, the least common ancestor or ancestors of the

21   set.

1   There are at least two ways to compute the LCA of the terms. One approach is to

2   store all non-leaf terms on the master server. This strategy is reasonably memory

3   efficient, since, in practice, most of the terms are leaves (minimal elements) in the partial

4   order. A second approach is to include the ancestors when returning the terms that are

5   refinements. This approach saves memory at the expense of increasing the size of the

6   data being transferred. The latter overhead is reasonable, since, in practice, a term

7   typically has very few ancestors.

8   The navigation system of the present invention allows information providers to

9   overlay a navigation system over any collection of documents. The knowledge base and

10   navigation aspects of the invention can be performed independently by different

11   providers, and information providers may outsource these functions to separate entities.

12   Similarly, a generated knowledge base may be imported by a navigation specialist.

13   Information providers may also outsource this navigation requirement to a navigation

14   system provider. A navigation system provider could charge customers a license fee for

15   the system independent of the amount of its usage. Alternatively, a navigation system

16   provider could charge customers on a per-click basis, a per-purchase basis if products are

17   available via the system, or per-transaction generated from a click through the navigation

18   system. A navigation system provider could also function as an aggregator -- compiling

19   records from a number of sources, combining them into a global data set, and generating

20   a navigation system to search the data set. The navigation system can be implemented as

21   software provided on a disk, on a CD, in memory, etc., or provided electronically (such

22   as over the Internet).

1   A navigation system in accordance with the present invention may also enhance

2   user profiling capability and merchandising capability. The navigation system may

3   maintain a profile of users based on the users' selections, including the particular paths

4   selected to explore the collection of navigation states. Using the knowledge base, the

5   system may also infer additional information regarding the users' preferences and

6   interests by supplementing the selection information with information regarding related

7   documents, attributes and terms in the knowledge base. That information may be used to

8   market goods and services related to the documents of interest to the user.

9   The foregoing description has been directed to specific embodiments of the

10   invention. The invention may be embodied in other specific forms without departing

11   from the spirit and scope of the invention. The embodiments, figures, terms and

12   examples used herein are intended by way of reference and illustration only and not by

13   way of limitation. The scope of the invention is indicated by the appended claims and all

14   changes that come within the meaning and scope of equivalency of the claims are

15   intended to be embraced therein.

16   We claim: